K Means Clustering
&
Hierarchical Clustering

AJITH M.S

Inclusys Org Foundation
Empowering the Neurodivergent

Agenda

Clustering.

Unsupervised Learning

K means Clustering

Random initialization trap

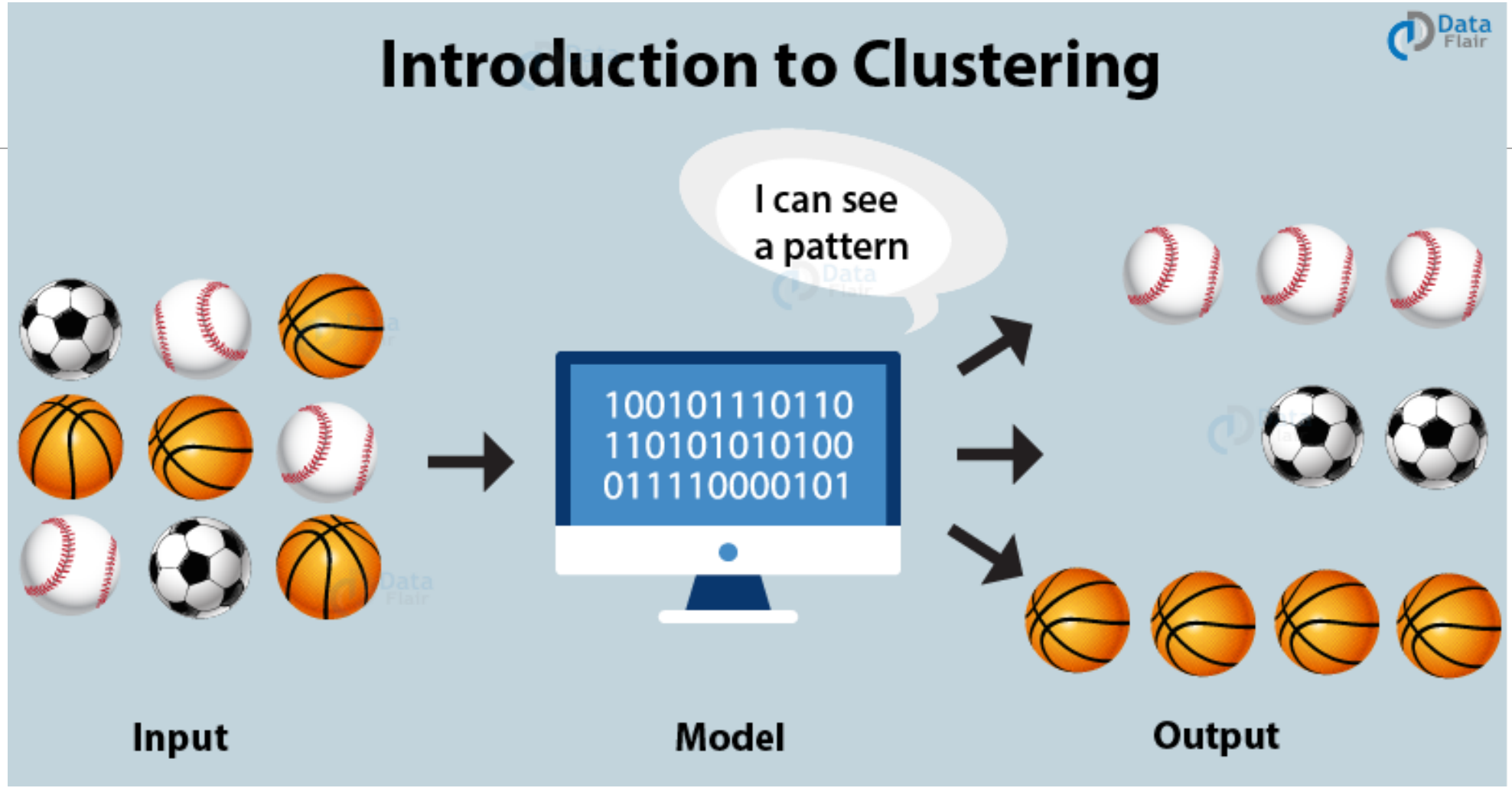
Elbow Method

Hierarchical Clustering

Clustering

- Clustering is a technique in machine learning and data analysis that involves grouping together similar data points or objects into subsets, called clusters.
- The goal of clustering is to identify patterns and structure in data, without requiring a specific outcome or target variable.
- The clustering process involves finding common characteristics among data points and organizing them into groups based on their similarity or proximity to each other.
- The similarity between data points is typically measured using a distance or similarity metric.
- There are many different types of clustering algorithms, including hierarchical clustering, k-means clustering, and density-based clustering. Clustering is used in a variety of applications, such as image and speech recognition, customer segmentation, anomaly detection, and bioinformatics.

- Clustering is a technique that involves grouping similar things together based on their common characteristics or proximity.
- In real-world applications, clustering can be used to identify patterns and structure in data, and to segment data into meaningful groups.
- This can help in tasks such as customer segmentation, anomaly detection, image recognition, and many other data analysis and machine learning applications.



Why Clustering in Machine Learning?

1. Unsupervised Learning: Clustering is a form of unsupervised learning, which means that it does not require labeled data or a specific outcome variable to be defined. Unsupervised learning algorithms can help identify patterns and relationships within the data without the need for prior knowledge or a specific objective.

2. Data Exploration: Clustering is an exploratory data analysis technique that can help uncover hidden patterns or structure within the data. It can provide insights into the underlying distribution of the data and identify groups or clusters of data points that share similar characteristics.

3. Feature Engineering: Clustering can also be used to create new features or variables that can be used in machine learning models. By clustering similar data points together, we can create a new variable that represents the cluster or group that the data point belongs to, which can be used as a feature in a predictive model.

4. Anomaly Detection: Clustering can be used to identify anomalous data points or outliers that do not fit within any of the identified clusters. This can help identify data quality issues or unusual behavior that may require further investigation.

Overall, clustering is a valuable tool in machine learning that can help identify patterns and structure in data, create new features, and support a variety of data analysis and modeling tasks.

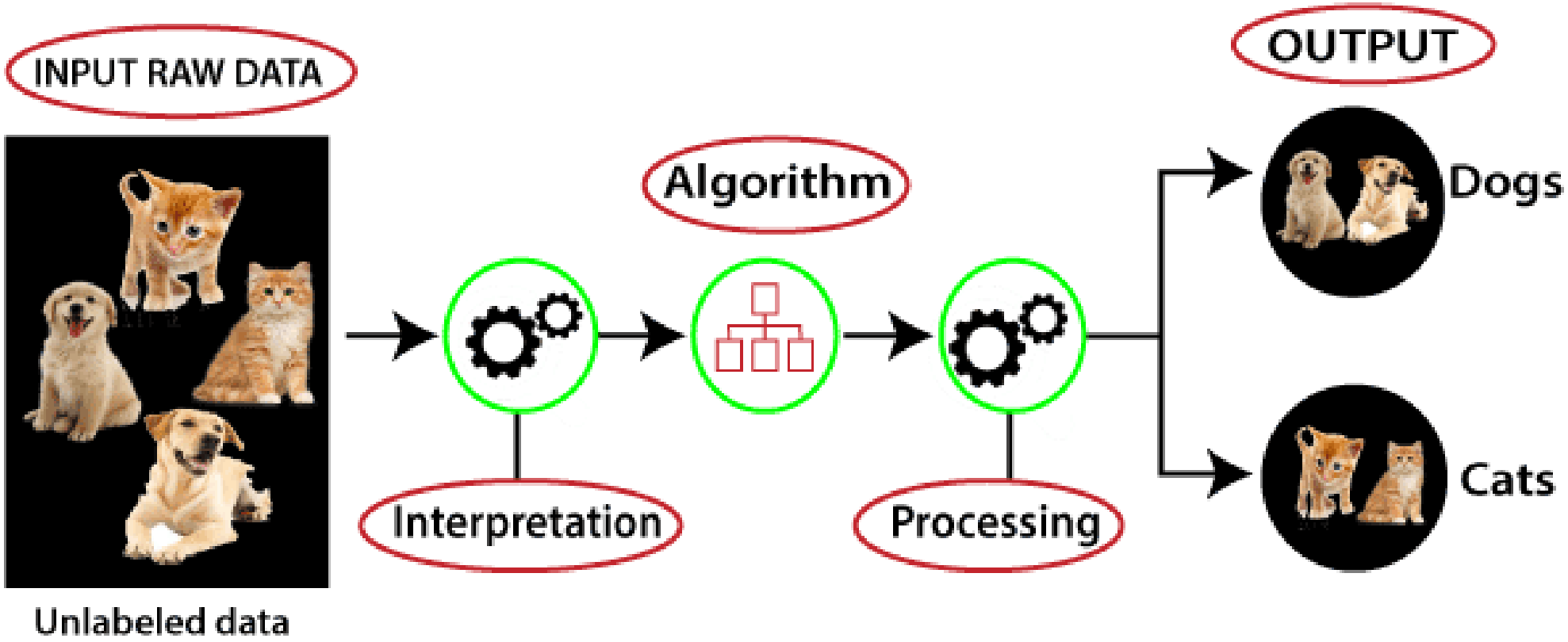
Unsupervised Learning

Unsupervised learning is a type of machine learning that involves analyzing and identifying patterns in data without the use of labeled examples or a specific outcome variable. In other words, the goal of unsupervised learning is to identify structure or relationships within the data itself.

As the name suggests, unsupervised learning is a machine learning technique in which models are not supervised using training dataset. Instead, models itself find the hidden patterns and insights from the given data. It can be compared to learning which takes place in the human brain while learning new things.

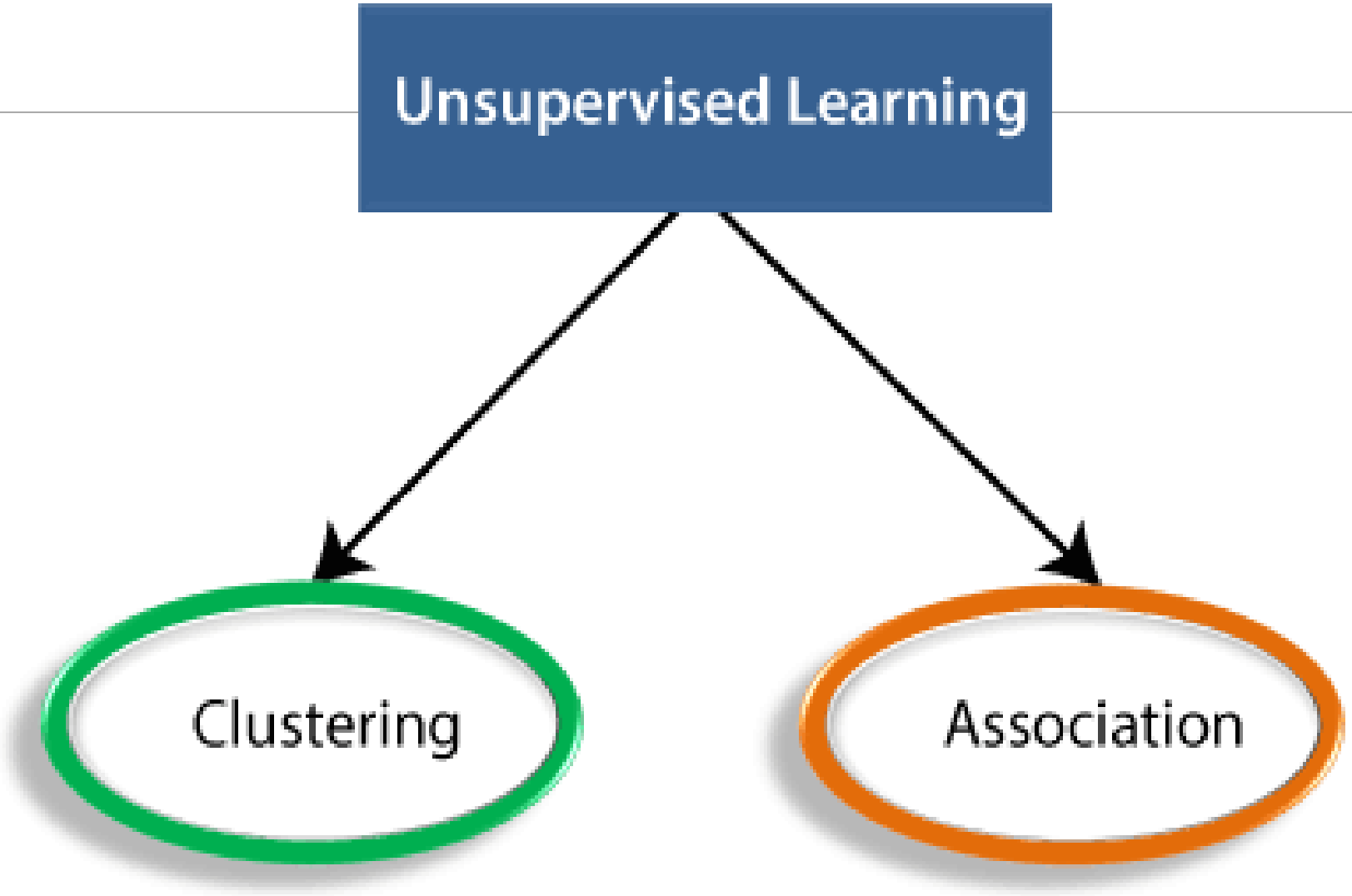
Example: Suppose the unsupervised learning algorithm is given an input dataset containing images of different types of cats and dogs. The algorithm is never trained upon the given dataset, which means it does not have any idea about the features of the dataset. The task of the unsupervised learning algorithm is to identify the image features on their own. Unsupervised learning algorithm will perform this task by clustering the image dataset into the groups according to similarities between images.

Working of unsupervised learning can be understood by the below diagram:



Here, we have taken an unlabeled input data, which means it is not categorized and corresponding outputs are also not given. Now, this unlabeled input data is fed to the machine learning model in order to train it. Firstly, it will interpret the raw data to find the hidden patterns from the data and then will apply suitable algorithms such as k-means clustering, Decision tree, etc.

Once it applies the suitable algorithm, the algorithm divides the data objects into groups according to the similarities and difference between the objects.



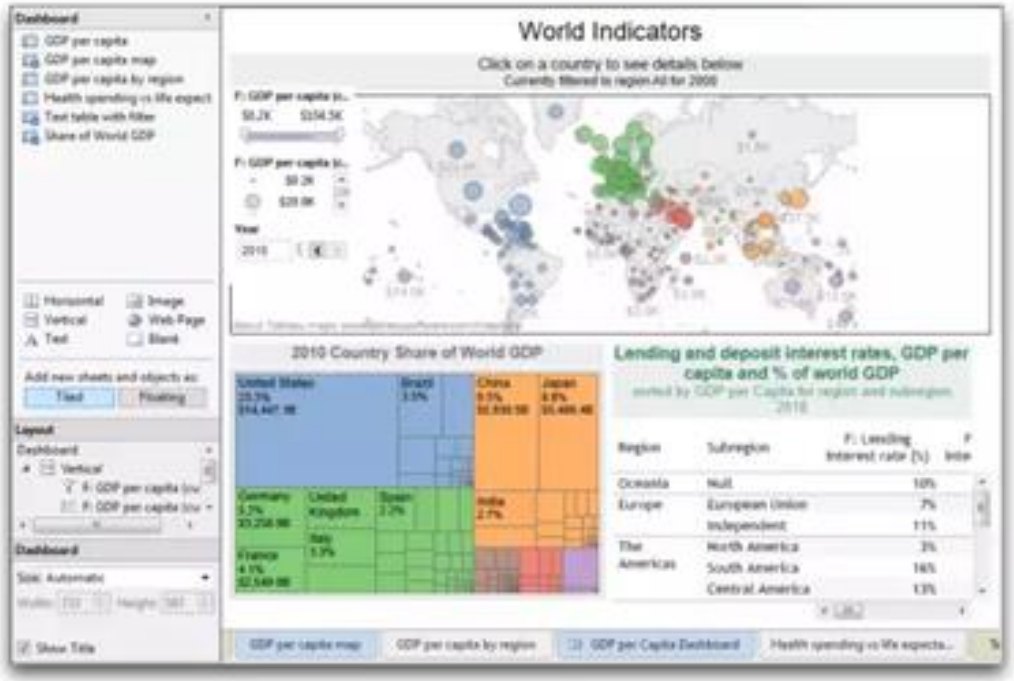
Association: An association rule is an unsupervised learning method which is used for finding the relationships between variables in the large database. It determines the set of items that occurs together in the dataset. Association rule makes marketing strategy more effective. Such as people who buy X item (suppose a bread) are also tend to purchase Y (Butter/Jam) item. A typical example of Association rule is Market Basket Analysis.

Inclusys Org Foundation
Empowering the Neurodivergent

Data Exploration

Data exploration is the first step in data analysis involving the use of data visualization tools and statistical techniques to uncover data set characteristics and initial patterns.

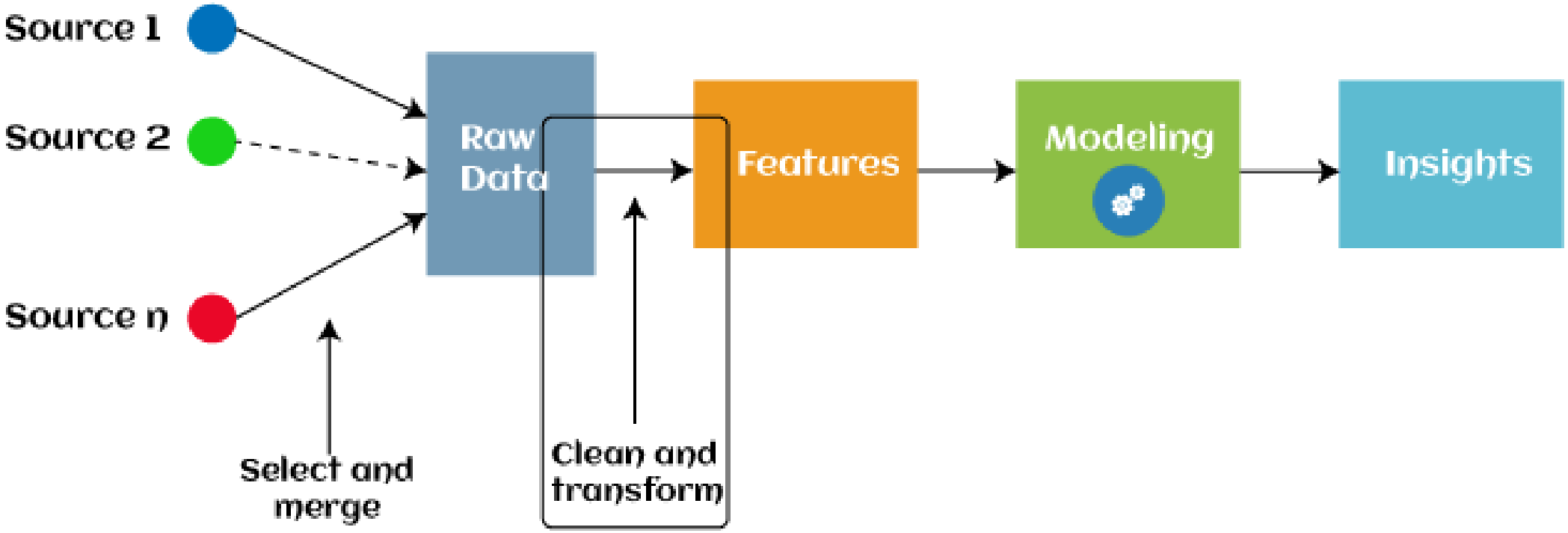
data exploration



data presentation

Feature Engineering

Feature engineering is the pre-processing step of machine learning, which extracts features from raw data. It helps to represent an underlying problem to predictive models in a better way, which as a result, improve the accuracy of the model for unseen data. The predictive model contains predictor variables and an outcome variable, and while the feature engineering process selects the most useful predictor variables for the model.



Anomaly Detection

Before talking about anomaly detection, we need to understand what an **anomaly** is.

Generally speaking, an anomaly is something that differs from a norm: a deviation, an exception. In software engineering, by anomaly we understand a rare occurrence or event that doesn't fit into the pattern, and, therefore, seems suspicious. Some examples are:

- sudden burst or decrease in activity;
- error in the text;
- sudden rapid drop or increase in temperature.

Common reasons for outliers are:

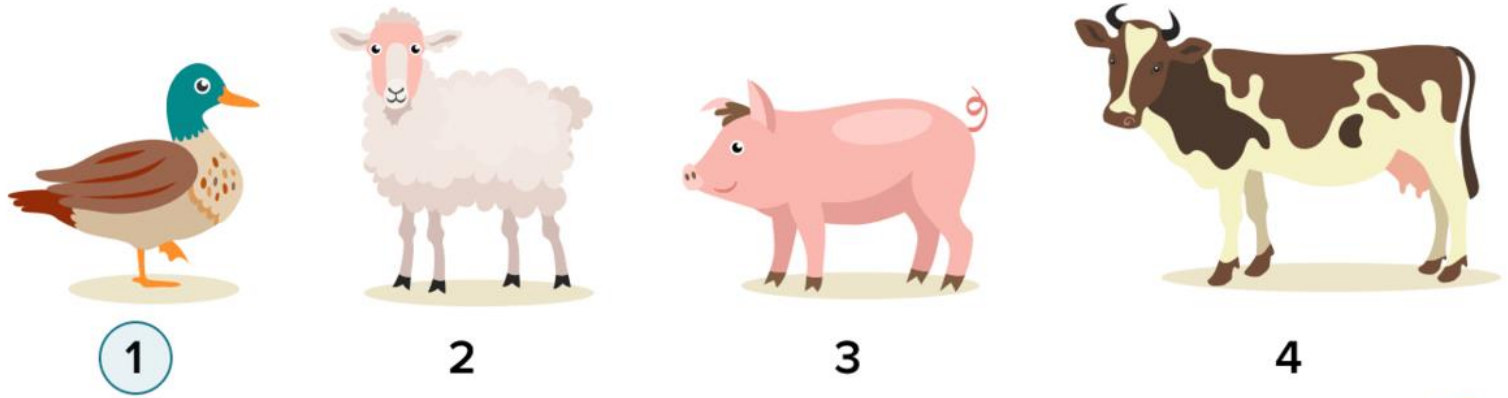
- data preprocessing errors;
- noise;
- fraud;
- attacks.

Normally, you want to catch them all; a software program must run smoothly and be predictable so every outlier is a potential threat to its robustness and security. Catching and identifying anomalies is what we call **anomaly or outlier detection**.

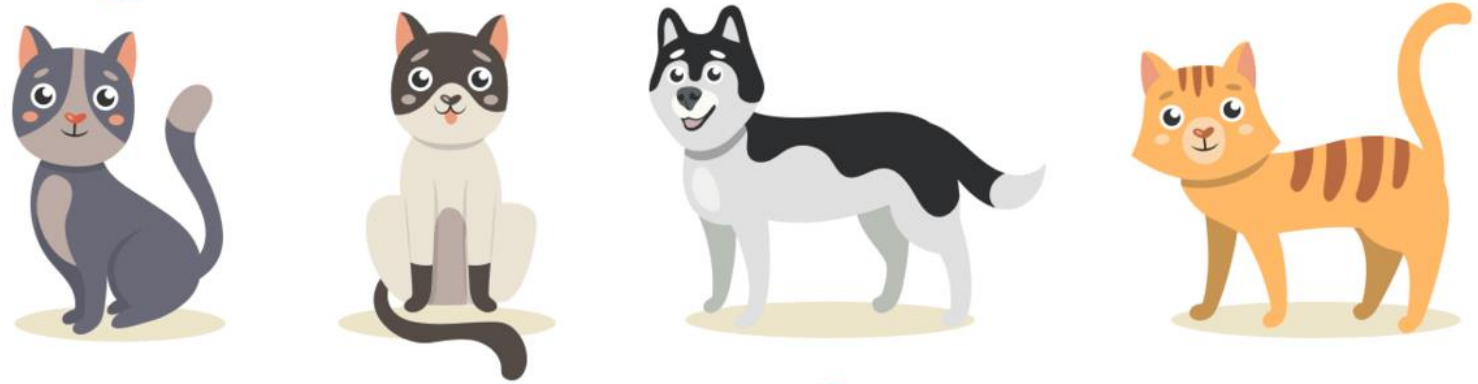
For example, if large sums of money are spent one after another within one day and it is not your typical behavior, a bank can block your card. They will see an unusual pattern in your daily transactions. This anomaly can typically be connected to fraud since identity thieves try to steal as much money as they can while they can. Once an anomaly is detected, it needs to be investigated, or problems may follow.

Inclusys Org Foundation
Empowering the Neurodivergent

Types of anomalies



1 2 3 4



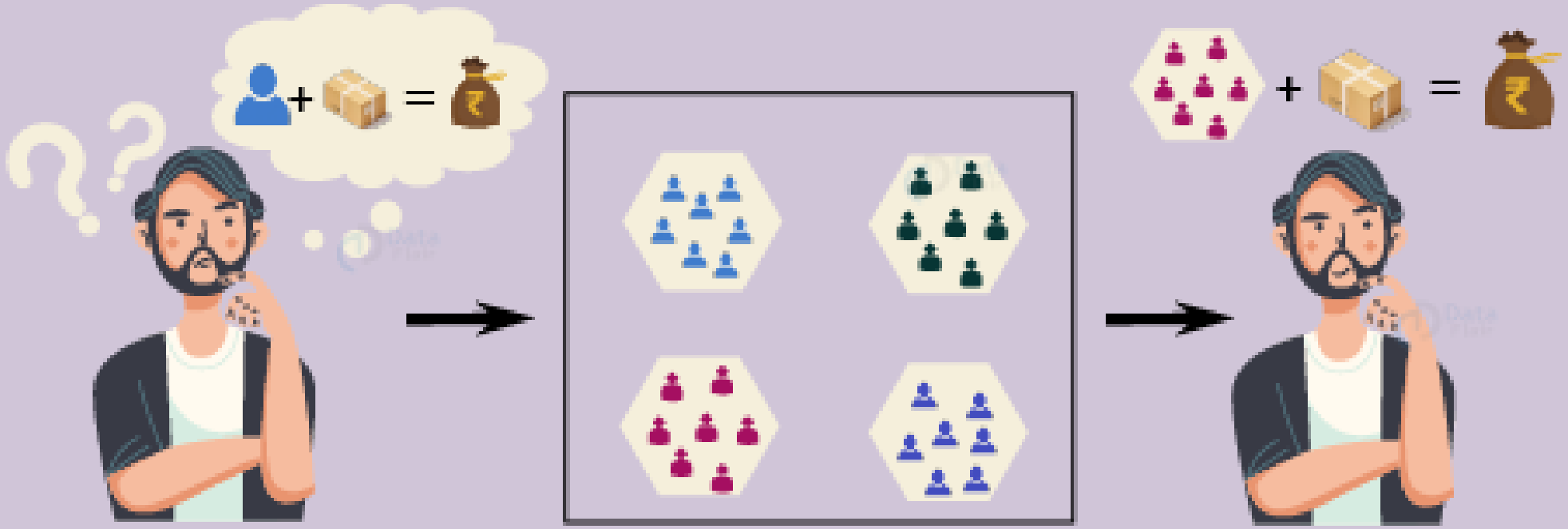
1 2 3 4

Types of Clustering Algorithms

In total, there are five distinct types of clustering algorithms. They are as follows –

- Partitioning Based Clustering
- Hierarchical Clustering
- Model-Based Clustering
- Density-Based Clustering
- Fuzzy Clustering

Clustering in Real World



Applications of Clustering

1. Clustering Algorithm for identification of cancer cells

Cancerous Datasets can be identified using clustering algorithms. In a mix of data consisting of both cancerous and non-cancerous data, the clustering algorithms are able to learn the various features present in the data upon which they produce the resulting clusters. Through experimentation, we observe that the cancerous data set gives us accurate results when given a model of unsupervised non-linear clustering algorithm.

2. Clustering Algorithm in Search Engines

While searching for something particular on Google, you receive a mix of similar results that match to your original query. This is a result of clustering that groups similar objects in a single cluster and provides that to you. Based on the nearest similar object, the data is assigned to the single cluster providing a comprehensive set of results to the user.

3. Clustering Algorithm in Wireless Networks

Using the clustering algorithm on the wireless nodes, we are able to save energy utilized by the wireless sensors. There are various clustering-based algorithms in wireless networks to improve their energy consumption and optimize data transmission

4. Clustering for Customer Segmentation

One of the most popular applications of clustering is in the field of customer segmentation. Based on the analysis of the user-base, companies are able to identify customers who would prove to be potential users for their product or services. Clustering allows them to segment customers into several clusters based on which they can adopt new strategies to appeal to their customer base. Now, you can practice the clustering concepts through the best ever machine learning project of the

K-Means CLUSTERING

K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science. In this topic, we will learn what is K-means clustering algorithm, how the algorithm works, along with the Python implementation of k-means clustering.

What is K-Means Algorithm?

K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.

"It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties."

It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.

It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

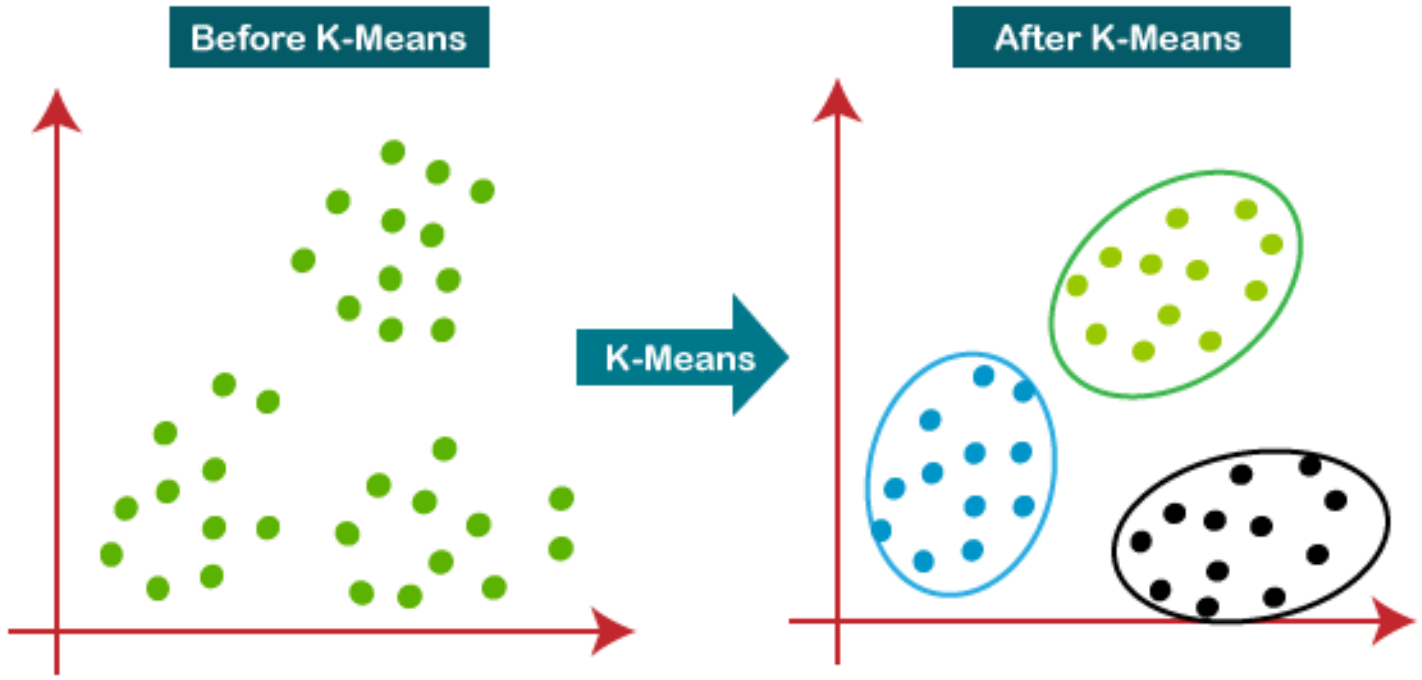
The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

The k-means clustering algorithm mainly performs two tasks:

- Determines the best value for K center points or centroids by an iterative process.
- Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

Hence each cluster has datapoints with some commonalities, and it is away from other clusters.

The below diagram explains the working of the K-means Clustering Algorithm:



How does the K-Means Algorithm Work?

The working of the K-Means algorithm is explained in the below steps:

Step-1: Select the number K to decide the number of clusters.

Step-2: Select random K points or centroids. (It can be other from the input dataset).

Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.

Step-4: Calculate the variance and place a new centroid of each cluster.

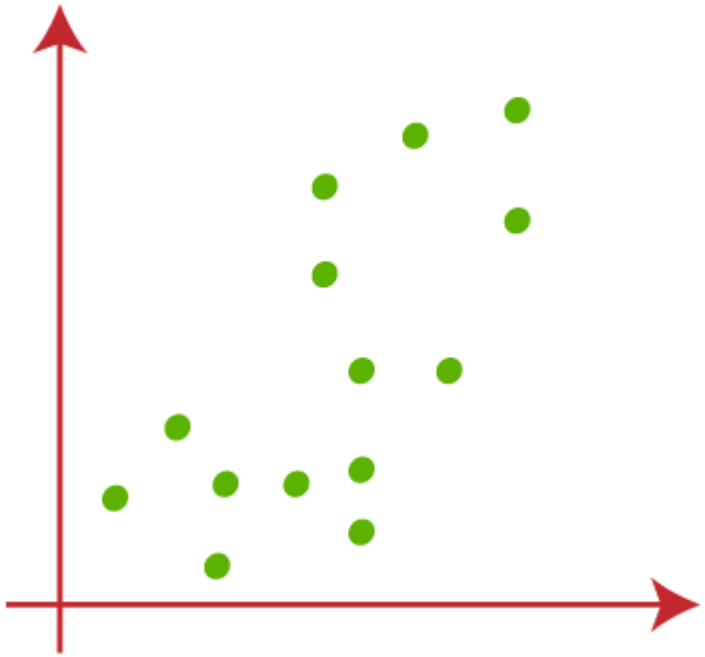
Step-5: Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.

Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.

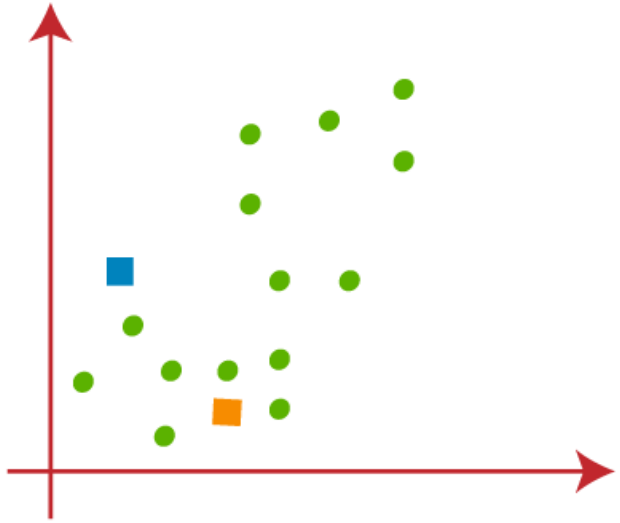
Step-7: The model is ready.

Let's understand the above steps by considering the visual plots:

Suppose we have two variables M1 and M2. The x-y axis scatter plot of these two variables is given below:

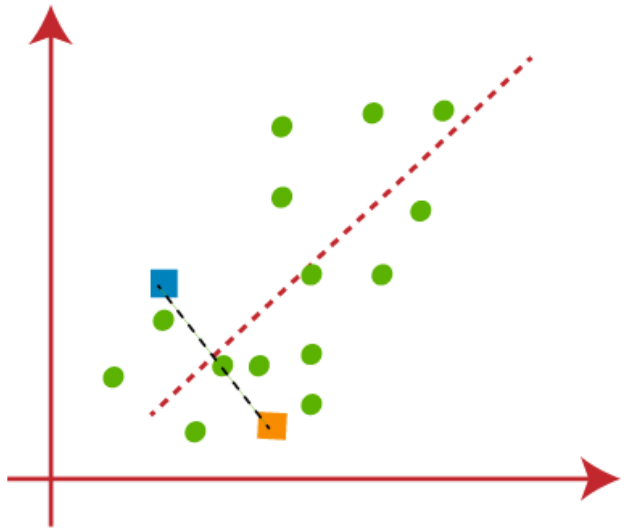


- Let's take number k of clusters, i.e., K=2, to identify the dataset and to put them into different clusters. It means here we will try to group these datasets into two different clusters.
- We need to choose some random k points or centroid to form the cluster. These points can be either the points from the dataset or any other point. So, here we are selecting the below two points as k points, which are not the part of our dataset. Consider the below image:

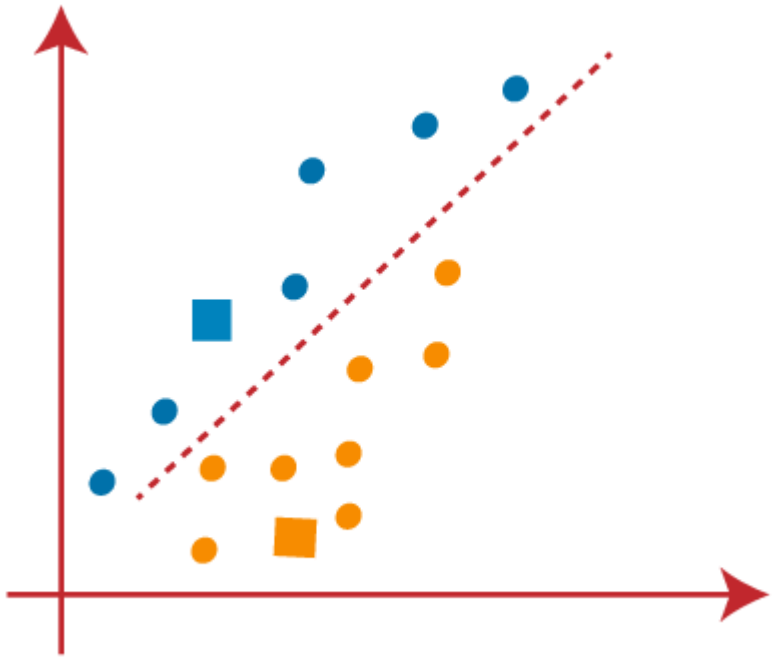


Inclusys Org Foundation
Empowering the Neurodivergent

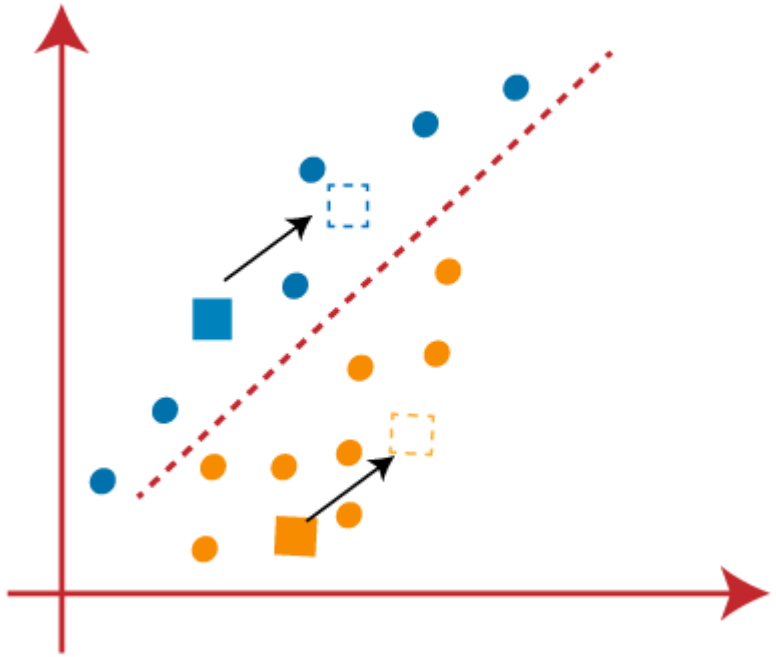
Now we will assign each data point of the scatter plot to its closest K-point or centroid. We will compute it by applying some mathematics that we have studied to calculate the distance between two points. So, we will draw a median between both the centroids. Consider the below image:



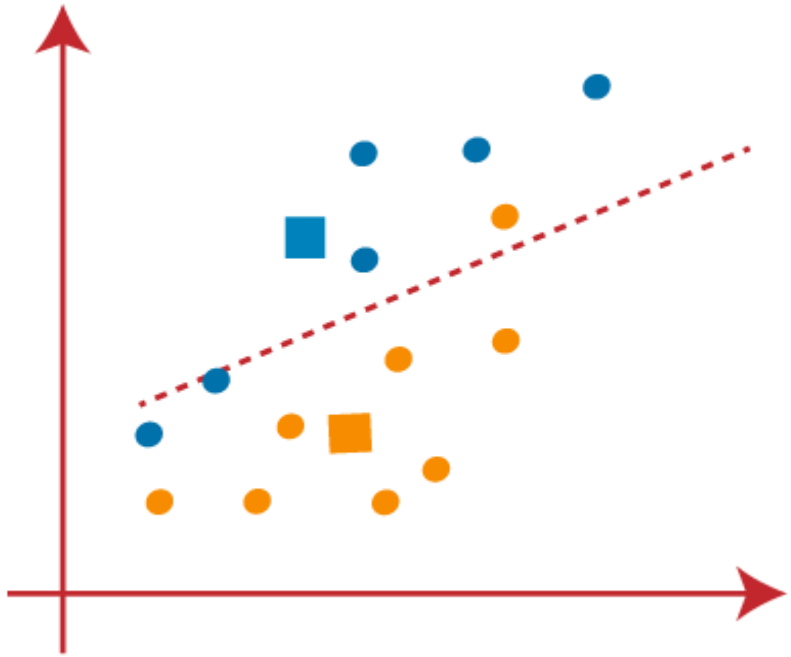
From the above image, it is clear that points left side of the line is near to the K1 or blue centroid, and points to the right of the line are close to the yellow centroid. Let's color them as blue and yellow for clear visualization.



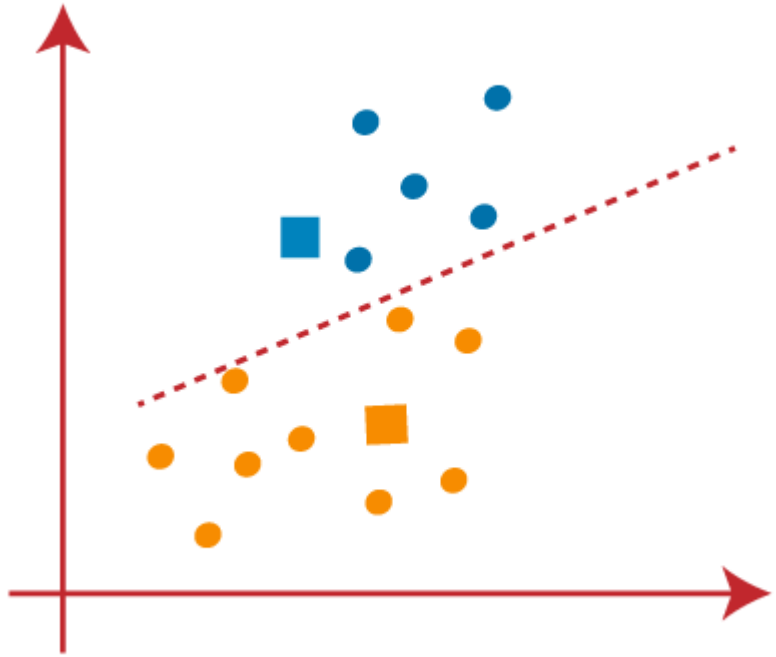
As we need to find the closest cluster, so we will repeat the process by choosing **a new centroid**. To choose the new centroids, we will compute the center of gravity of these centroids, and will find new centroids as below:



Next, we will reassign each datapoint to the new centroid. For this, we will repeat the same process of finding a median line. The median will be like below image:

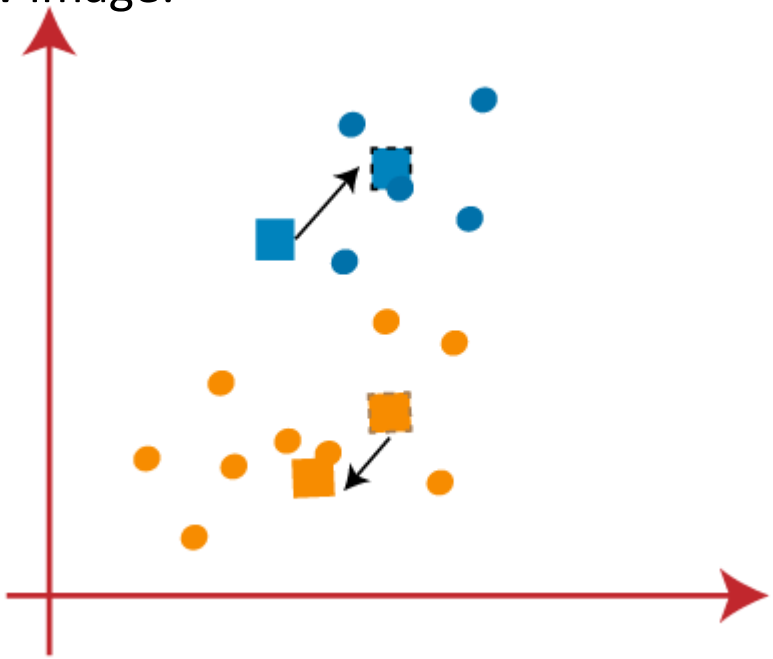


From the above image, we can see, one yellow point is on the left side of the line, and two blue points are right to the line. So, these three points will be assigned to new centroids.

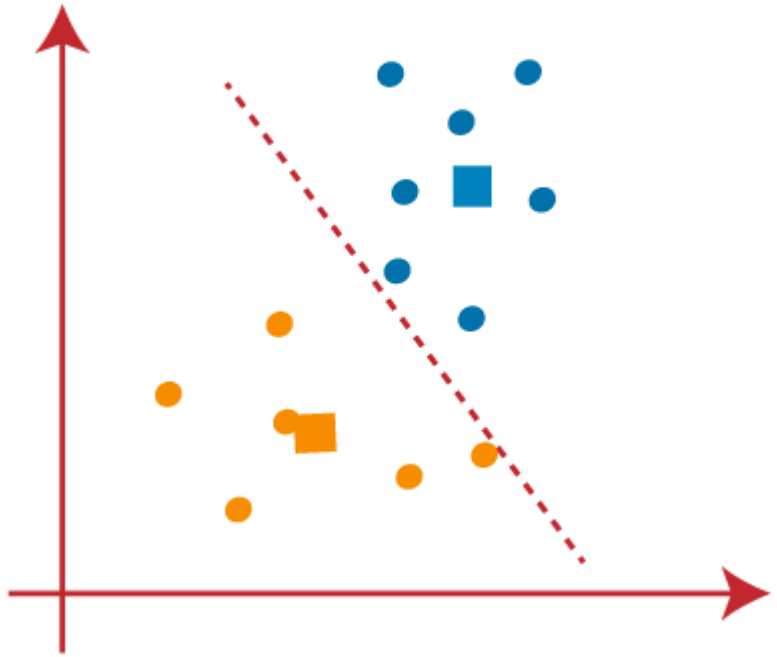


As reassignment has taken place, so we will again go to the step-4, which is finding new centroids or K-points.

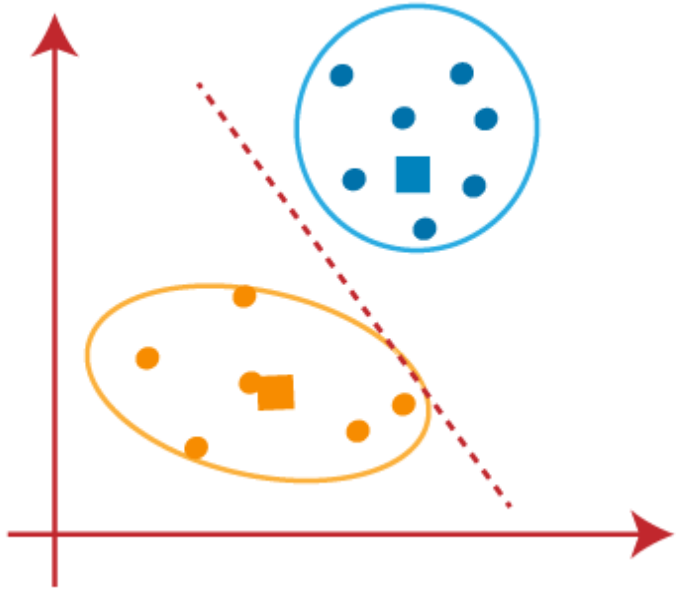
- We will repeat the process by finding the center of gravity of centroids, so the new centroids will be as shown in the below image:



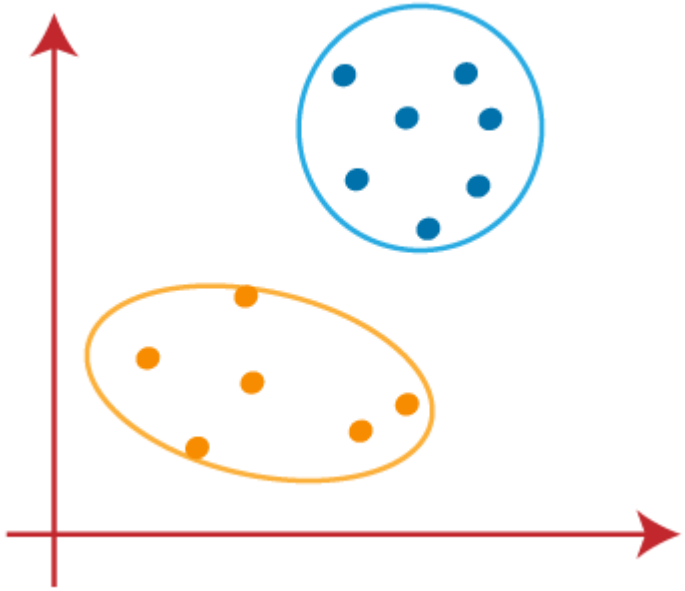
As we got the new centroids so again will draw the median line and reassign the data points. So, the image will be:



We can see in the above image; there are no dissimilar data points on either side of the line, which means our model is formed. Consider the below image:



As our model is ready, so we can now remove the assumed centroids, and the two final clusters will be as shown in the below image:

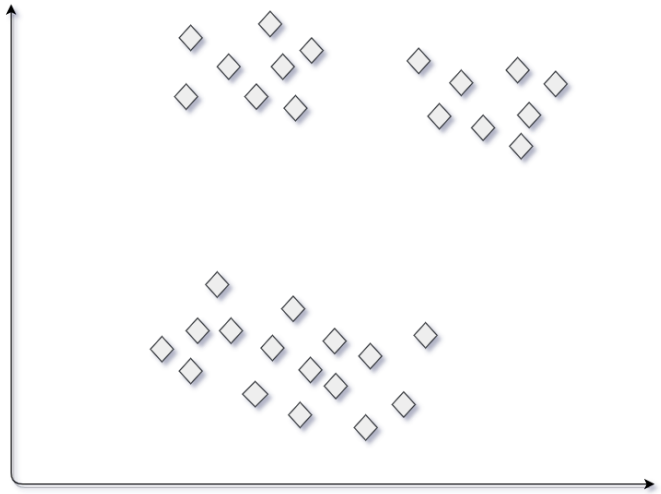


Random Initialization Trap in K-Means

Inclusys Org Foundation
Empowering the Neurodivergent

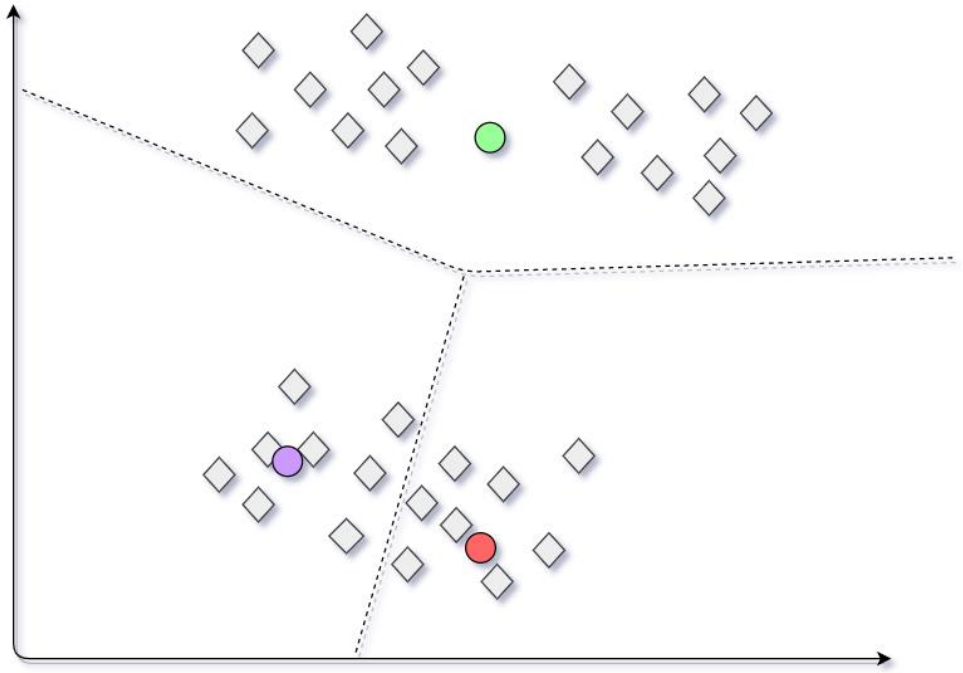
Random initialization trap is a problem that occurs in the K-means algorithm. In random initialization trap when the centroids of the clusters to be generated are explicitly defined by the User then inconsistency may be created and this may sometimes lead to generating wrong clusters in the dataset. So random initialization trap may sometimes prevent us from developing the correct clusters.

Example :
Suppose you have a dataset with the following points shown in the picture and you want to generate three clusters in this dataset according to their attributes by performing K-means clustering. From the figure, we can get the intuition what are the clusters that are required to be generated. K-means will perform clustering on the basis of the centroids fed into the algorithm and generate the required clusters according to these centroids.



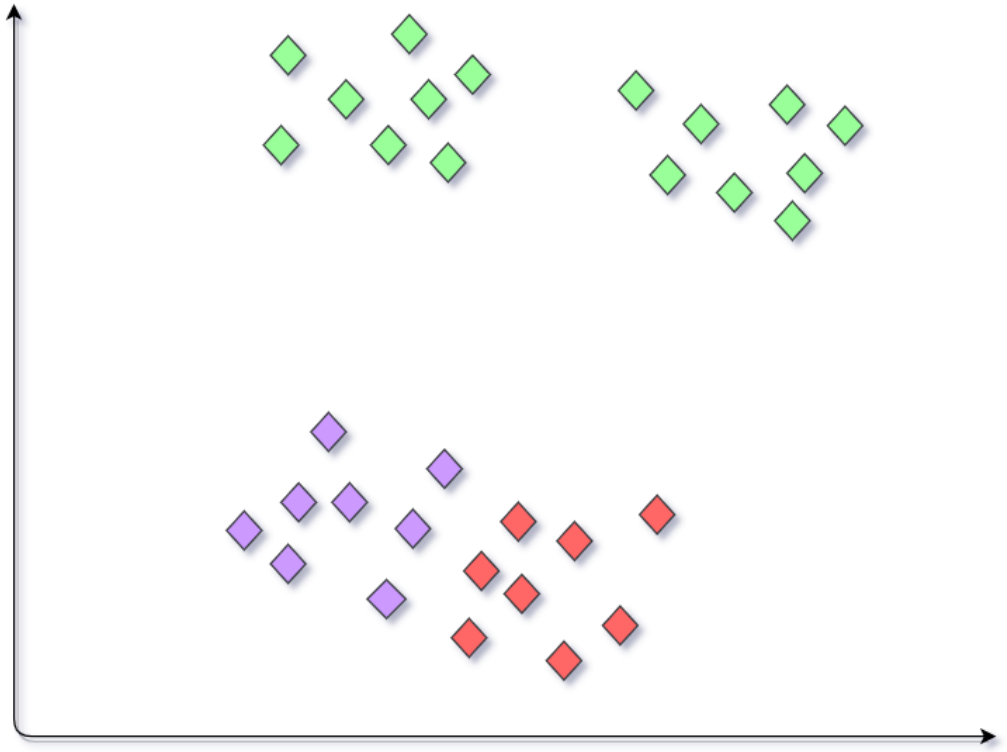
First Trial

Suppose we choose 3 sets of centroids according to the figure shown below. The clusters that are generated corresponding to these centroids are shown in the figure below.



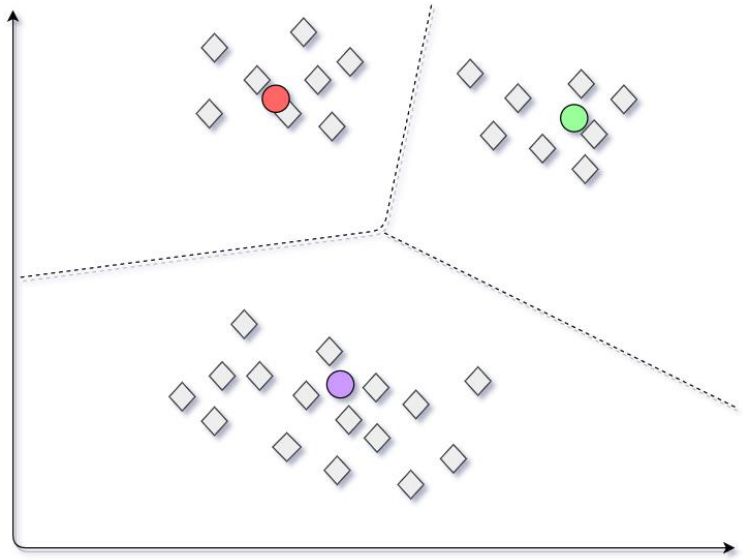
Inclusys Org Foundation
Empowering the Neurodivergent

Final Model.

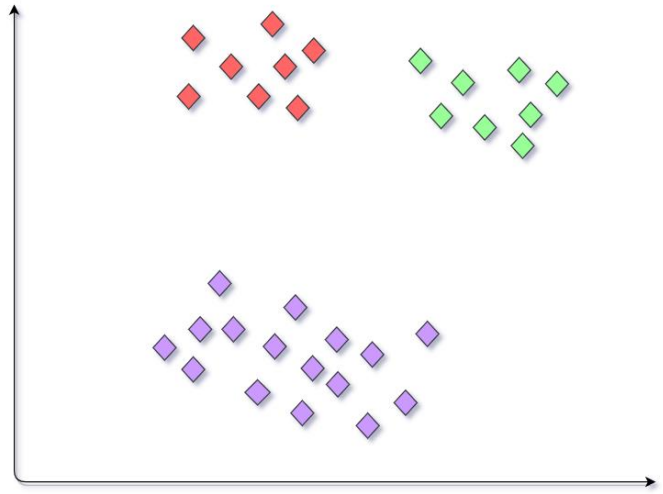


Second Trial

Consider another case in which we choose another set of centroids for the dataset as shown. Now the set of clusters generated will be different from the clusters generated in the previous practice.



Final Model



Similarly we may get different model outputs on the same dataset. This condition where a different set of clusters is generated when a different set of centroids are provided to the K-means algorithm making it inconsistent and unreliable is called the Random initialization trap.

How to choose the value of "K number of clusters" in K-means Clustering?

The performance of the K-means clustering algorithm depends upon highly efficient clusters that it forms. But choosing the optimal number of clusters is a big task. There are some different ways to find the optimal number of clusters, but here we are discussing the most appropriate method to find the number of clusters or value of K. The method is given below:

Elbow Method

The Elbow method is one of the most popular ways to find the optimal number of clusters. This method uses the concept of WCSS value. **WCSS** stands for **Within Cluster Sum of Squares**, which defines the total variations within a cluster. The formula to calculate the value of WCSS (for 3 clusters) is given below:

$$WCSS= \sum_{Pi \text{ in Cluster1}} distance(P_i\ C_1)^2 + \sum_{Pi \text{ in Cluster2}} distance(P_i\ C_2)^2 + \sum_{Pi \text{ in CLuster3}} distance(P_i\ C_3)^2$$

In the above formula of WCSS,

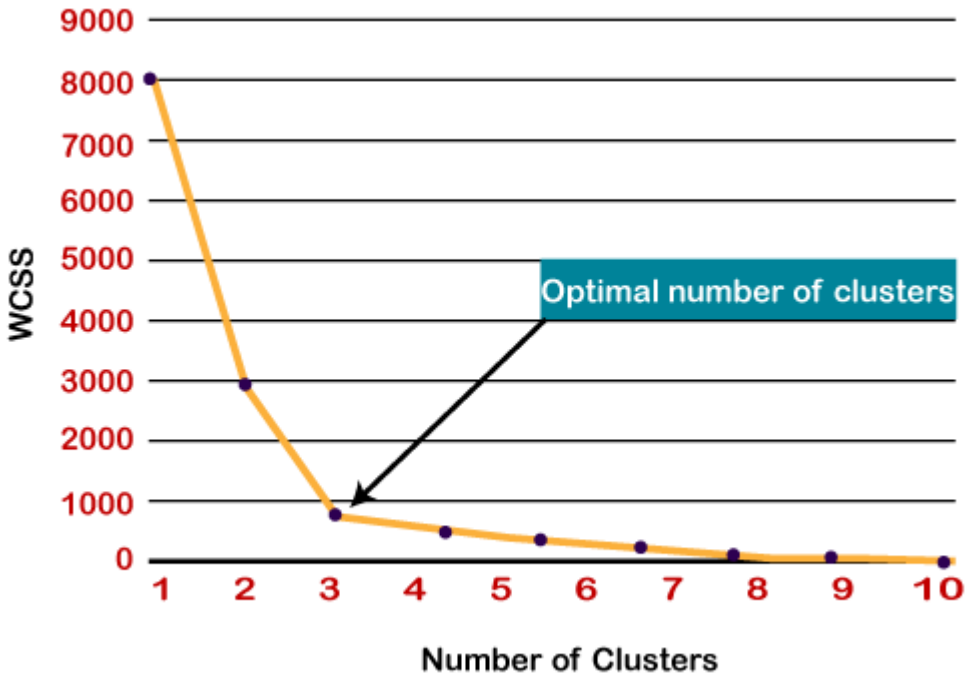
$\sum_{Pi \text{ in Cluster1}}$ distance(P_i C_1)^2: It is the sum of the square of the distances between each data point and its centroid within a cluster1 and the same for the other two terms.

To measure the distance between data points and centroid, we can use any method such as Euclidean distance or Manhattan distance.

To find the optimal value of clusters, the elbow method follows the below steps:

- It executes the K-means clustering on a given dataset for different K values (ranges from 1-10).
- For each value of K, calculates the WCSS value.
- Plots a curve between calculated WCSS values and the number of clusters K.
- The sharp point of bend or a point of the plot looks like an arm, then that point is considered as the best value of K.

Since the graph shows the sharp bend, which looks like an elbow, hence it is known as the elbow method. The graph for the elbow method looks like the below image:



Note: We can choose the number of clusters equal to the given data points. If we choose the number of clusters equal to the data points, then the value of WCSS becomes zero, and that will be the endpoint of the plot.

Hierarchical Clustering in Machine Learning

Inclusys Org Foundation
Empowering the Neurodivergent

Hierarchical clustering is another unsupervised machine learning algorithm, which is used to group the unlabeled datasets into a cluster and also known as **hierarchical cluster analysis** or HCA.

In this algorithm, we develop the hierarchy of clusters in the form of a tree, and this tree-shaped structure is known as the **dendrogram**.

Sometimes the results of K-means clustering and hierarchical clustering may look similar, but they both differ depending on how they work. As there is no requirement to predetermine the number of clusters as we did in the K-Means algorithm.

What is Hierarchical Clustering?

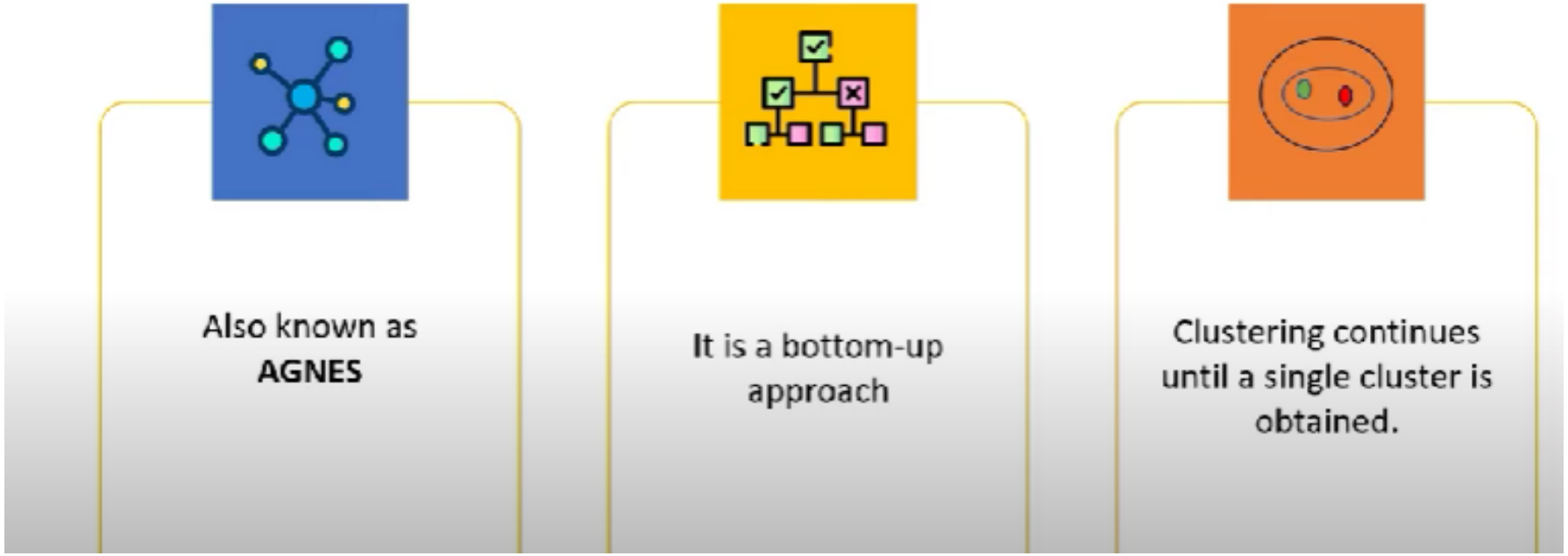
The hierarchical clustering technique has two approaches:

1. **Agglomerative:** Agglomerative is a **bottom-up** approach, in which the algorithm starts with taking all data points as single clusters and merging them until one cluster is left.
2. **Divisive:** Divisive algorithm is the reverse of the agglomerative algorithm as it is a **top-down approach.**

Agglomerative Hierarchical clustering

The agglomerative hierarchical clustering algorithm is a popular example of HCA. To group the datasets into clusters, it follows the **bottom-up approach**. It means, this algorithm considers each dataset as a single cluster at the beginning, and then start combining the closest pair of clusters together. It does this until all the clusters are merged into a single cluster that contains all the datasets.

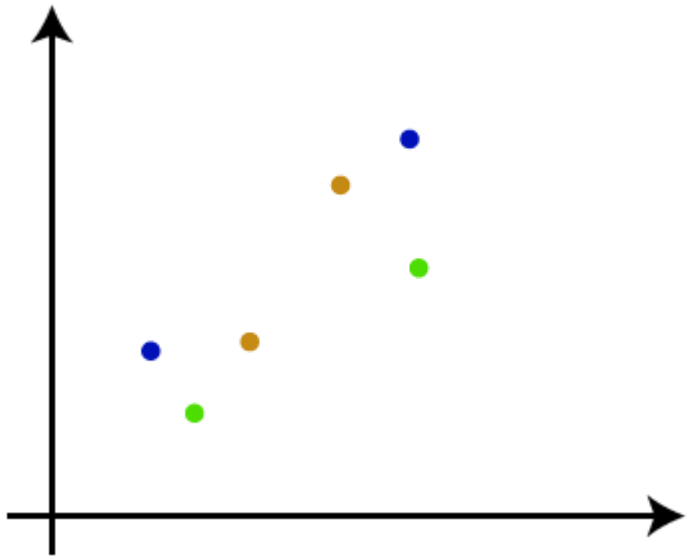
This hierarchy of clusters is represented in the form of the dendrogram.


Also known as
AGNES
It is a bottom-up approach
Clustering continues until a single cluster is obtained.

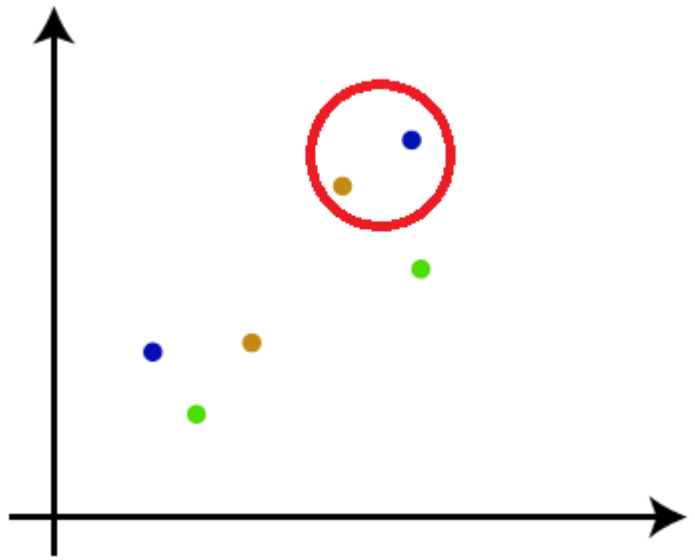
How the Agglomerative Hierarchical clustering Work?

Inclusys Org Foundation
Empowering the Neurodivergent

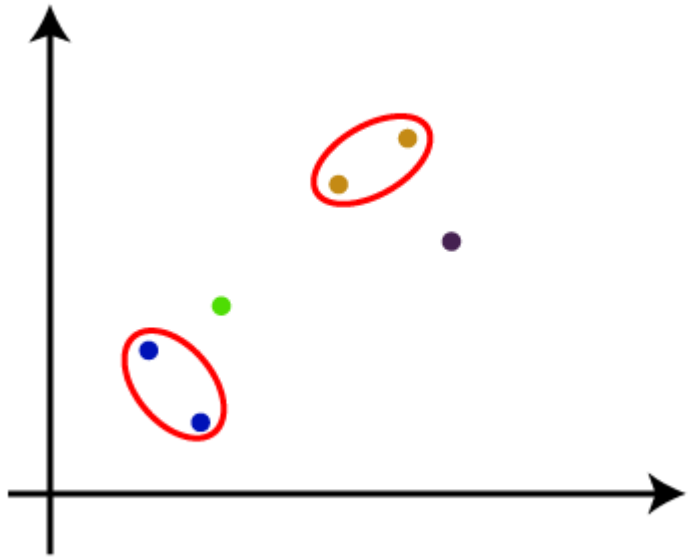
Step-1: Create each data point as a single cluster. Let's say there are N data points, so the number of clusters will also be N.



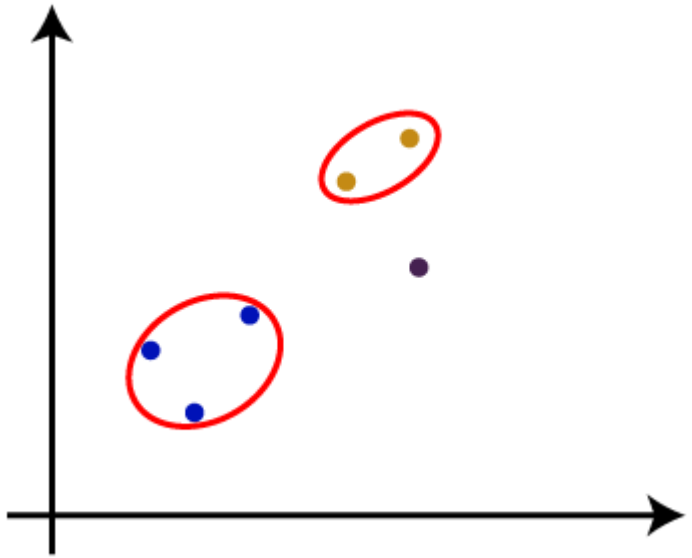
Step-2: Take two closest data points or clusters and merge them to form one cluster. So, there will now be N-1 clusters.

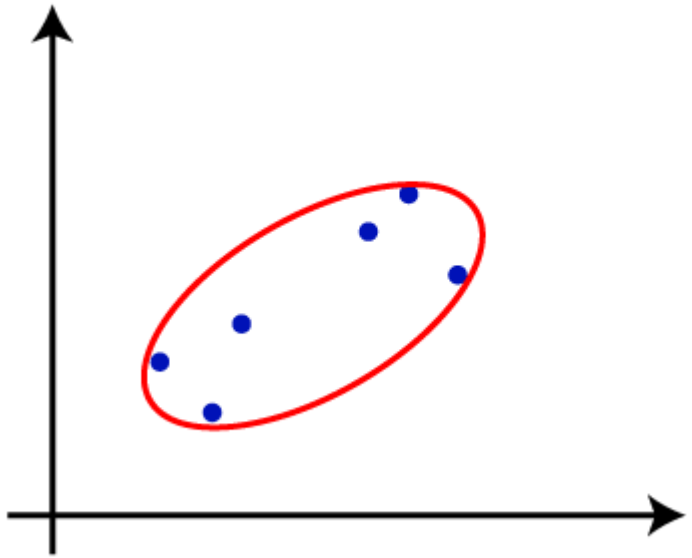


Step-3: Again, take the two closest clusters and merge them together to form one cluster. There will be N-2 clusters.



Step-4: Repeat Step 3 until only one cluster left. So, we will get the following clusters. Consider the below images:



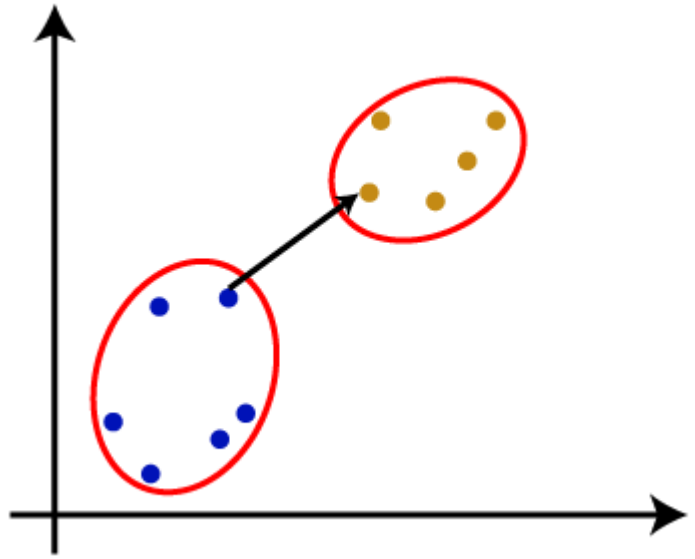


Step-5: Once all the clusters are combined into one big cluster, develop the dendrogram to divide the clusters as per the problem.

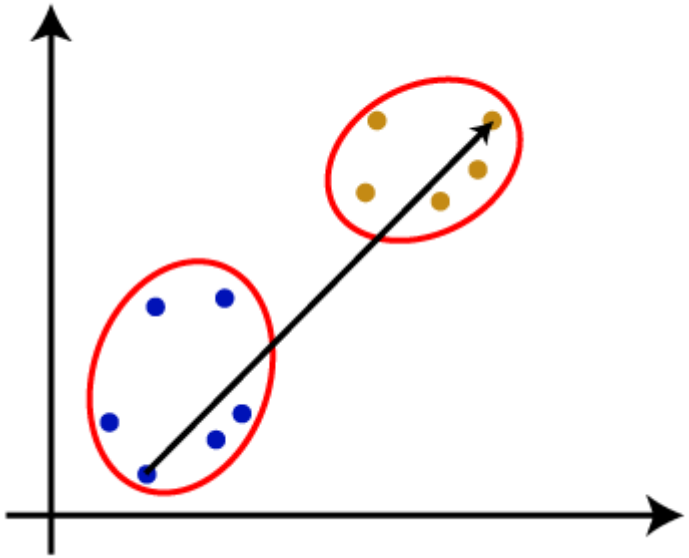
Measure for the distance between two clusters

As we have seen, the **closest distance** between the two clusters is crucial for the hierarchical clustering. There are various ways to calculate the distance between two clusters, and these ways decide the rule for clustering. These measures are called **Linkage methods**. Some of the popular linkage methods are given below:

Single Linkage: It is the Shortest Distance between the closest points of the clusters. Consider the below image:

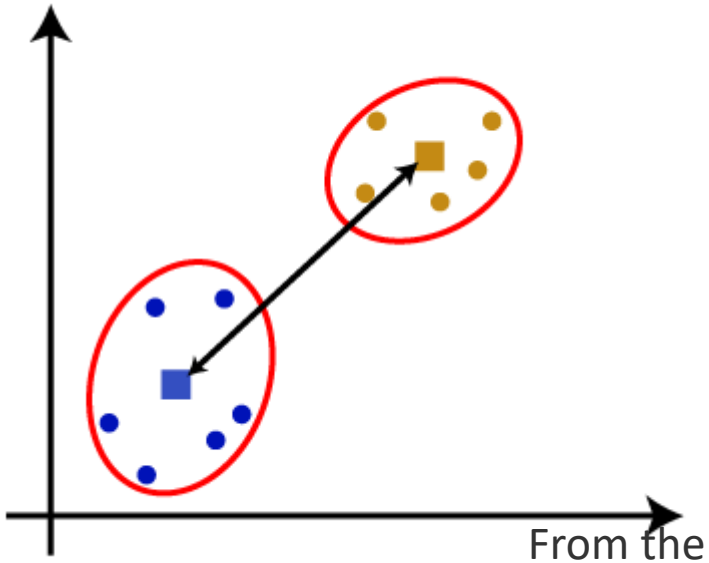


Complete Linkage: It is the farthest distance between the two points of two different clusters. It is one of the popular linkage methods as it forms tighter clusters than single-linkage.



Average Linkage: It is the linkage method in which the distance between each pair of datasets is added up and then divided by the total number of datasets to calculate the average distance between two clusters. It is also one of the most popular linkage methods.

Centroid Linkage: It is the linkage method in which the distance between the centroid of the clusters is calculated. Consider the below image:

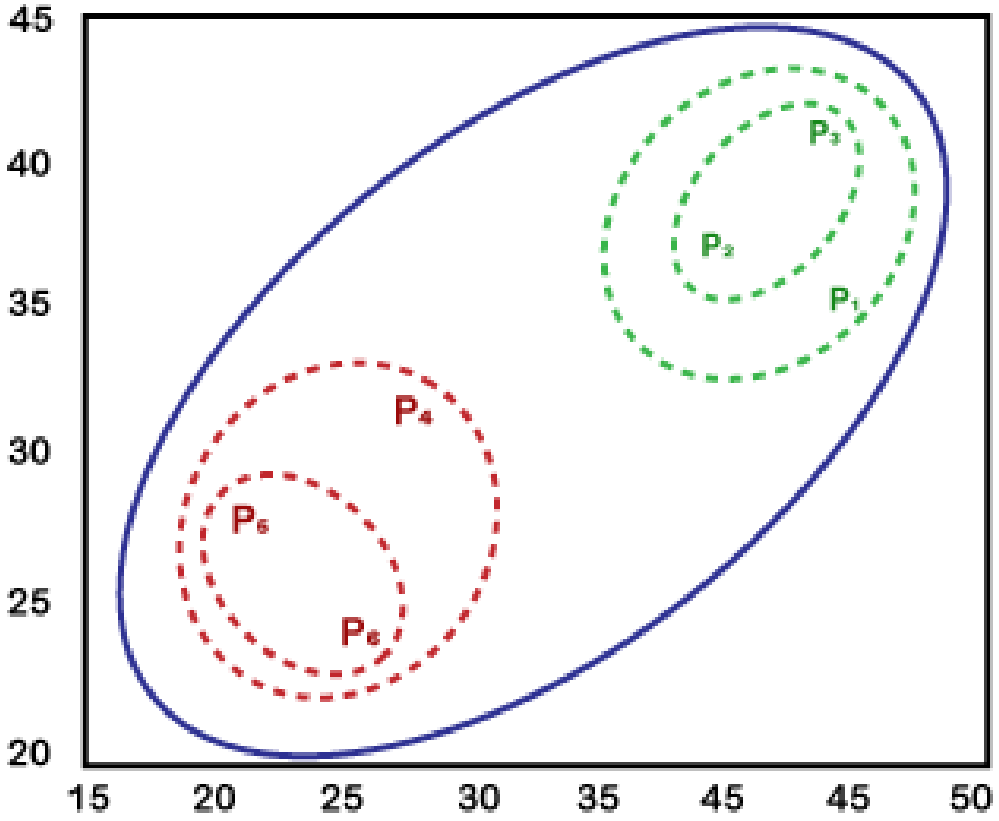


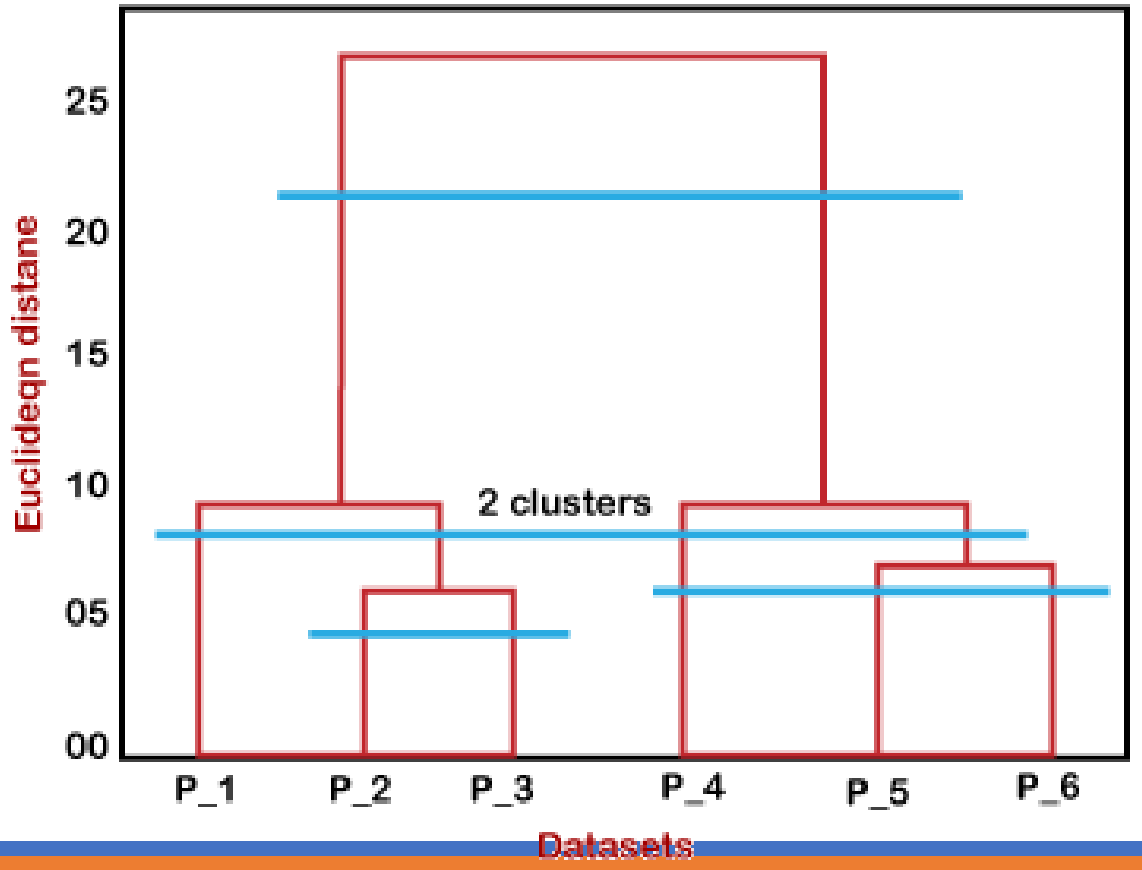
From the above-given approaches, we can apply any of them according to the type of problem or business requirement.

Working of Dendrogram in Hierarchical clustering

The dendrogram is a tree-like structure that is mainly used to store each step as a memory that the HC algorithm performs. In the dendrogram plot, the Y-axis shows the Euclidean distances between the data points, and the x-axis shows all the data points of the given dataset.

The working of the dendrogram can be explained using the below diagram:





In the above diagram, the left part is showing how clusters are created in agglomerative clustering, and the right part is showing the corresponding dendrogram.

- As we have discussed above, firstly, the datapoints P2 and P3 combine together and form a cluster, correspondingly a dendrogram is created, which connects P2 and P3 with a rectangular shape. The hight is decided according to the Euclidean distance between the data points.
- In the next step, P5 and P6 form a cluster, and the corresponding dendrogram is created. It is higher than of previous, as the Euclidean distance between P5 and P6 is a little bit greater than the P2 and P3.
- Again, two new dendrograms are created that combine P1, P2, and P3 in one dendrogram, and P4, P5, and P6, in another dendrogram.
- At last, the final dendrogram is created that combines all the data points together.

We can cut the dendrogram tree structure at any level as per our requirement.

Problem

Consider the following set of 6 one dimensional datapoints:

18,22,25,42,27,43

We need to apply Agglomerative Hierarchical Clustering algorithm to build the hierarchical clustering dendogram

➔ Merge the clusters using min distance and update the proximity matrix accordingly

➔ Clearly show the proximity matrix corresponding to each iteration of the algoritham

- Step – 1

	18	22	25	27	42	43
18	0	4	7	9	24	25
22	4	0	3	5	20	21
25	7	3	0	2	17	18
27	9	5	2	0	15	16
42	24	20	17	15	0	1
43	25	21	18	16	1	0

- Step – 1

	18	22	25	27	42	43
18	0	4	7	9	24	25
22	4	0	3	5	20	21
25	7	3	0	2	17	18
27	9	5	2	0	15	16
42	24	20	17	15	0	1
43	25	21	18	16	1	0

(42, 43)

- Step – 2

	18	22	25	27	42, 43
18	0	4	7	9	24
22	4	0	3	5	20
25	7	3	0	2	17
27	9	5	2	0	15
42, 43	24	20	17	15	0

- Step – 2

	18	22	25	27	42, 43
18	0	4	7	9	24
22	4	0	3	5	20
25	7	3	0	2	17
27	9	5	2	0	15
42, 43	24	20	17	15	0

~~(42, 43), (25, 27)~~

- Step – 3

	18	22	25, 27	42, 43
18	0	4	7	24
22	4	0	3	20
25, 27	7	3	0	15
42, 43	24	20	15	0

- Step – 3

	18	22	25, 27	42, 43
18	0	4	7	24
22	4	0	3	20
25, 27	7	3	0	15
42, 43	24	20	15	0

(42, 43), ((25, 27), 22)

- Step – 4

	18	22, 25, 27	42, 43
18	0	4	24
22, 25, 27	4	0	15
42, 43	24	15	0

- Step – 4

	18	22, 25, 27	42, 43
18	0	4	24
22, 25, 27	4	0	15
42, 43	24	15	0

(42, 43), (((25, 27), 22), 18)

- Step – 5

	18, 22, 25, 27	42, 43
18, 22, 25, 27	0	15
42, 43	15	0

- Step – 5

	18, 22, 25, 27	42, 43
18, 22, 25, 27	0	15
42, 43	15	0

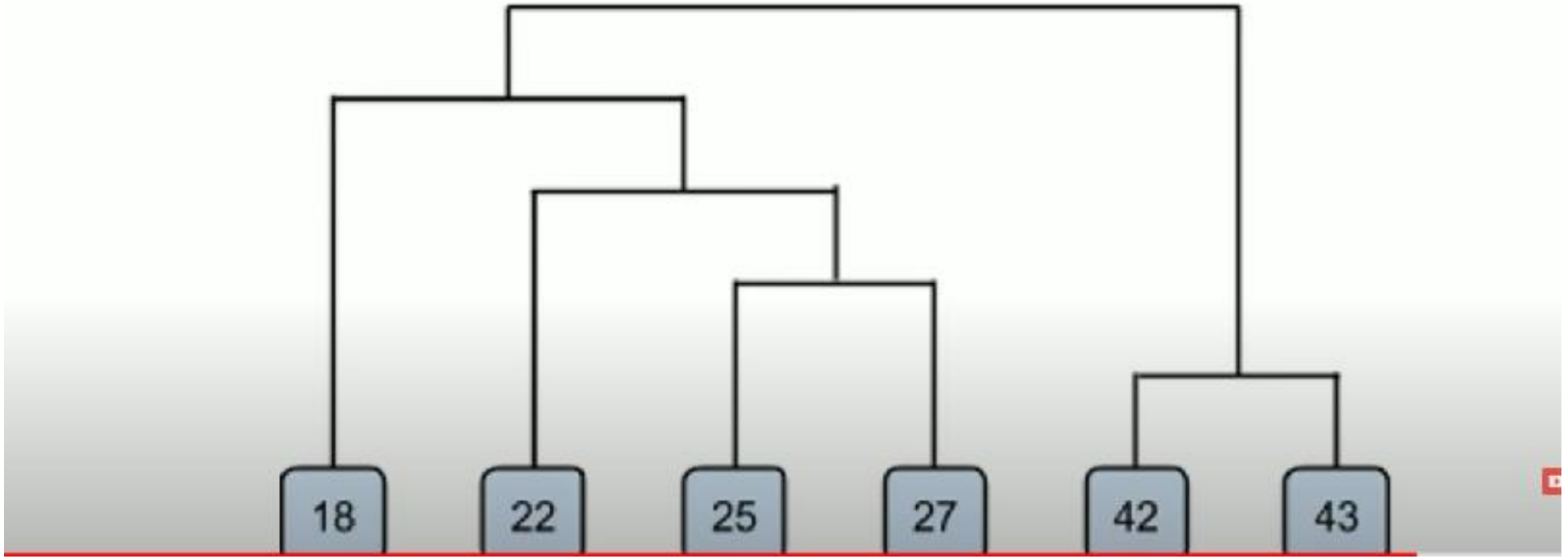
((42, 43), (((25, 27), 22), 18))

- Step – 6

	18, 22, 25, 27, 42, 43
18, 22, 25, 27, 42, 43	0

- **Dendrogram**

((42, 43), (((25, 27), 22), 18))



Inclusys Org Foundation
Empowering the Neurodivergent

THANK YOU